

# Myszka komputerowa dla osób niepełnosprawnych, część 1

## AVT-862

**PROJEKT Z OKŁADKI**



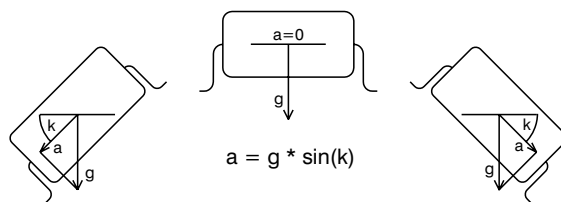
Przedstawione poniżej urządzenie ma dość nietypową konstrukcję. Zawiera dwa nowoczesne akcelerometry i czujnik ciśnienia z precyzyjnym przetwornikiem analogowo-cyfrowym. Sygnały z tych sensorów przetwarza szybki procesor o architekturze RISC.

To wszystko służy... przesuwaniu kursora na ekranie komputera.

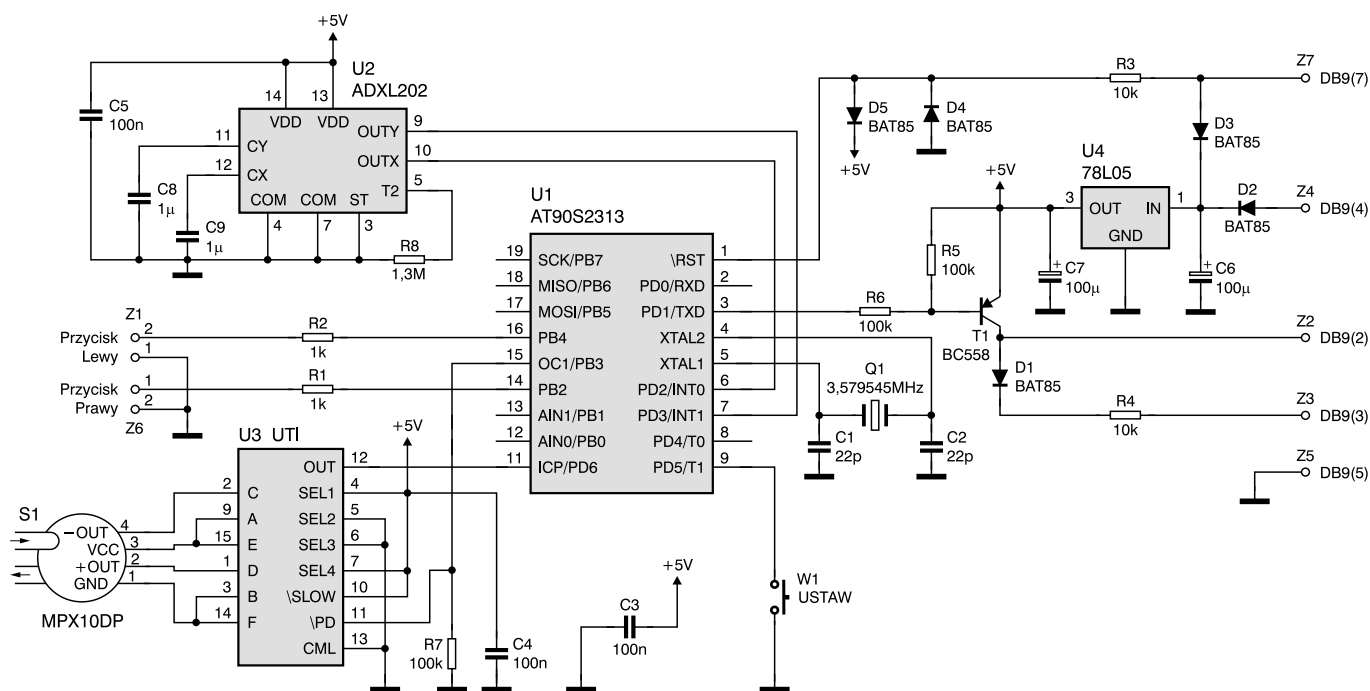
Czy wykorzystanie tak zaawansowanych technologii jest konieczne do budowy tytułowej myszki komputerowej? Tak, jeśli założymy, że myszka ma umożliwić dostęp do komputera osobie, która może pracować tylko głową. I to dosłownie. Opisane poniżej urządzenie jest bowiem przeznaczone dla osób z niedowładem rąk i nóg. Zdarza się, że takie schorzenia są wynikiem choroby i z tym musimy się pogodzić. Niestety, często wiąże się to z urazami, które powstają w czasie wypadków komunikacyjnych lub lekkomyślnych zabaw na przykład nad wodą. Wiele, zbyt wiele ofiar tych wy-

padków, to ludzie młodzi mający przed sobą wiele lat życia i to życia, które wcale nie musi być wegetacją.

Radio, telewizja, telefonia stacjonarna, a ostatnio komórkowa, komputery i Internet - to wszystko ma służyć zaspokajaniu naszych potrzeb. Wiele osiągnięć techniki jest dla nas, ludzi zdrowych, tylko kolejnym udogodnieniem. Dla osób niepełnosprawnych niektóre z tych mediów mogą stać się prawdziwym dobrodziejstwem. Ktoś, kto nie spędził całych miesięcy unieruchomiony w łóżku, gdzie jedyną dostępną rozrywką jest książka lub ogłupiająca telewizja, nie jest w stanie wyobrazić sobie, czym dla osoby sparaliżowanej, może stać się komputer i dostęp do Internetu. Nie, nie tylko źródłem rozrywki. Na internetowych grupach dyskusyjnych taka osoba może rozmawiać. Strony www są niewyczerpanym źródłem wiedzy. Dzięki telepracy, osoba niepełnosprawna może



Rys. 1. Wpływ przechyłu na wartość przyspieszenia ziemskiego widzianego przez akcelerometr.



Rys. 2. Schemat elektryczny myszki.

znaleźć zatrudnienie. Nie twierdę, że to wszystko jest łatwe, ale jestem przekonany, że to jest możliwe.

Do pokonania istniejących barier nie zawsze są potrzebne marmurowe podjazdy dla wózków inwalidzkich. Czasem wystarczą niepozorne rzeczy. Mam nadzieję, że przedstawiona myszka stanie się dla kogoś takim pomocnym drobiazgiem.

Na początek proponuję prosty eksperyment, który uzmysłowi nam, na jakie problemy napotykają osoby z niedowładem rąk. Usiądź sobie wygodnie przed komputerem, dłonie połóż na kolanach. Teraz napisz na komputerze swoje imię. Gdzie te łapy? Z powrotem na kolana! Możesz wziąć w usta ołówek i spróbować za jego pomocą naciskać klawisze. Jeśli nie znajdziesz na biurku ołówka, pozostaje pisanie nosem. To nie jest żart. Gdy już się podpiszesz, spróbuj poruszyć myszką i kliknąć! Ciężko, nieprawdaż? A teraz pomyśl, że i tak jesteś w dobrej sytuacji, bo poruszasz swobodnie tułowiem, czego osoby sparaliżowane zwykle robić nie mogą.

Konstruując urządzenie, staram się myśleć o przyszłym użytkowniku, o jego potrzebach i możliwościach. Wy, drodzy Czytelnicy, budując różnego rodzaju urządze-

nia zazwyczaj widzicie siebie jako ich przyszłych użytkowników. Nie sądzę, aby wśród Czytelników EP znalazło się wiele osób dotkniętych tetraplegią. Natomiast nie mam wątpliwości, że niektórzy z Was znają osoby, którym myszka mogłaby się przydać.

Ze skrucich muszę przyznać, że przez te kilka lat odkąd mam okazję pisywać dla Elektroniki Praktycznej, nie zaprojektowałem żadnego układu z myślą o osobach niepełnosprawnych. Niewielką pociechą jest fakt, że z czystym sumieniem mógłbym grzotnąć w pierś prawie całe grono redakcyjne. Oszczędziłbym jedynie Jurka Szczesiula, który stworzył program klawiatury wirtualnej. Program ten zostanie przedstawiony w jednym z kolejnych numerów EP. W skrócie powiem tylko, że dzięki programowi wirtualnej klawiatury można, posługując się jedynie myszką, szybko i wygodnie wpisywać teksty do komputera. A jeśli do komputera będzie podłączona opracowana przeze mnie myszka, to naprawdę wystarczy do tego tylko zdolność do wykonywania niewielkich przechyłów głowy oraz umiejętność dmuchnięcia i zassania powietrza przez ustnik.

Najważniejsze parametry myszki przedstawiono w **tab. 1**. Przyjrzyjmy się im bliżej. Nasza

myszka jest widziana przez komputer jako standardowa myszka pracująca w systemie Microsoft Serial Mouse. Dzięki temu nie wymaga specjalnych sterowników i pracuje poprawnie w środowisku graficznym Windows i programach działających w systemie DOS.

Elektronika myszki mieści się w niewielkiej obudowie wielkości paczki papierosów. Do poprawnej pracy konieczne jest umieszczenie myszki na głowie osoby obsługującej. Przechylenie głowy do przodu i do tyłu powoduje na ekranie przemieszczanie kursora na dół i do góry. Analogiczne ruchy głowy w lewo i prawo powodują zmianę pozycji kursora w osi X.

Wyszedłem z założenia, że projektowana myszka powinna umożliwić pracę osobie całkowicie sparaliżowanej, która może poruszać jedynie głową. Dlatego myszka jest zaopatrzona w czujnik pneumatyczny z ustnikiem. Dmuchięcie w rurkę jest widziane przez komputer jako naciśnięcie lewego klawisza myszy. Zassanie powietrza odpowiada naciśnięciu prawego klawisza.

Niezależnie od tego, do myszki można podłączyć dwa dodatkowe wyłączniki. Dzięki temu do naciśnięcia klawiszy można wykorzystać ruchy np. władnego palca.

**Tab. 1. Podstawowe parametry myszki.**

- ✓ praca w standardzie MS-MOUSE;
- ✓ zmiana pozycji kursora przez przechylenie głowy;
- ✓ przełącznik pneumatyczny:
  - ✗ dmuchnięcie - klawisz lewy;
  - ✗ zassanie - klawisz prawy;
- ✓ możliwość podłączenia dodatkowych wyłączników;
- ✓ zasilanie z portu szeregowego;
- ✓ pobór prądu około 6mA.

Możliwości adaptacji jest wiele i kilka propozycji zostanie przedstawionych na końcu artykułu.

Niewielki pobór prądu umożliwia zasilanie myszki z portu szeregowego. Takie rozwiązanie jest wygodne i powszechnie stosowane.

### Założenia projektu

Jak zwykle w takich przypadkach, proces konstruowania rozpocząłem od przeglądu urządzeń oferowanych na rynku i spełniających podobne funkcje. Mimo dość intensywnych poszukiwań nie udało mi się znaleźć zbyt wielu gotowych rozwiązań.

W jednym z najciekawszych, jakie znalazłem, zastosowano reflektor oświetlający twarz osoby siedzącej przed monitorem światłem z zakresu podczerwieni. Obraz z umieszczonej obok miniaturowej kamery był przeszukiwany pod kątem najjaśniejszego punktu. Ruchy tego punktu były przenoszone na kursor. Wielką zaletą takiego rozwiązania jest to, że operator nie jest w żaden sposób „uwiązany“ do komputera. Wystarczy umieścić gdzieś na twarzy niewielką, odblaskową naklejkę. Więcej szczegółów na temat tego rozwiązania można znaleźć na internetowej stronie [www.orin.com/access/headmouse](http://www.orin.com/access/headmouse).

Pozostałe rozwiązania bazowały na miniaturowych żyroskopach i bardziej przypominały hełmy do oglądania wirtualnej rzeczywistości, niż proste urządzenia wskazujące. Wyjątkiem była bezprzewodowa gyro-mouse, którą można obejrzeć na stronie [www.advancedperipheral.com/gyro.htm](http://www.advancedperipheral.com/gyro.htm).

Cechą wspólną gotowych rozwiązań jest nieproporcjonalnie wysoka cena, częstokroć przekraczająca cenę dobrej klasy komputera.

Początkowo założyłem, że ruch kursora w poziomie będzie wymu-

szany przez kręcenie głowy w lewo i w prawo. Pochylenie głowy do tyłu i do przodu miało przesuwać kursor odpowiednio w górę i w dół.

W pierwszym odruchu zamierzałem skonstruować myszkę w oparciu o żyroskopy. Znalazłem nawet miniaturowe żyroskopy elektromagnetyczne, które do działania wykorzystują efekt Coriolisa. Firma Gyration ([www.gyration.com](http://www.gyration.com)) zmieściła dwa takie żyroskopy w obudowie o wymiarach 25x25x20mm. Byłem zdecydowany zastosować te ciekawe elementy, niestety, nie do pokonania okazała się bariera cenowa. Dodatkowo zniechęcała mnie konieczność stosowania specjalizowanych układów do odczytywania sygnałów wyjściowych.

Potem wymyśliłem, żeby do mierzenia pochylenia głowy w kierunku przód-tył wykorzystać czujnik przyspieszenia pracujący jako poziomnica (patrz EP9/99). Teraz pozostało już tylko wykrywanie kręcenia głową. Potencjalnie nadawałby się do tego czujnik pola magnetycznego. Wymaga on jednak oddalenia od źródeł zakłóceń elektromagnetycznych i precyzyjnego wypoziomowania. Okolice komputera trudno nazwać obszarem o niskim poziomie zakłóceń, a wypoziomowanie trudno pogodzić z koniecznością pochylenia głowy.

W sumie musiałem zrezygnować z wykrywania kręcenia głową, które wydawało się bardziej naturalne, na rzecz przechylania, które jest łatwiejsze do mierzenia. Dzięki przyjęciu takiego rozwiązania, mogłem w obu płaszczyznach zastosować stosunkowo tanie akcelerometry.

Mój wybór padł na układ ADXL202 firmy Analog Devices. W ceramicznej, 14-nóżkowej obudowie do montażu powierzchniowego, znajdują się dwa ustawione prostopadle akcelerometry. Układ wymaga do pracy niewielu elementów zewnętrznych, a dzięki wyjściom PWM doskonale nadaje się do współpracy z mikrokontrolerem. Dodatkową zaletą jest standardowe zasilanie (4,75..5,25V) i niewielki pobór prądu (<1mA). Jednym słowem wymarzony element do naszej myszki.

Akcelerometry zawarte w układzie ADXL202 mogą mierzyć przyspieszenia w zakresie  $\pm 2g$ . Czułość przetwarzania jest stała i wynosi 12,5%/g z tolerancją  $\pm 2,5\%$ . Oznacza to, że współczynnik wypełnienia prostokątnego przebiegu wyjściowego zmienia się o około 12,5% przy zmianie przyspieszenia o 9,81m/s<sup>2</sup>.

Przy idealnie poziomym ustawieniu akcelerometru, przyspieszenie ziemskie przez niego mierzone wynosi zero. Przy odchyleniu przyrządu od poziomu o określony kąt, mierzona wartość przyspieszenia będzie rosła zgodnie z wartością sinusa kąta odchylenia. Ilustruje to **rys. 1**. Dla 90 stopni sinus osiąga wartość jeden i w takim przypadku mierzone przyspieszenie wyniesie 1g.

Teraz wyobraźmy sobie akcelerometr leżący na głowie. Gdy trzymamy głowę prosto, akcelerometr leży w przybliżeniu poziomo. Przechylając głowę powodujemy odchylenie czujnika od poziomu w jedną lub drugą stronę. Bez większych trudności można przechylić głowę na boki o 45 stopni, a w przód i w tył nawet więcej. Tyle tylko, że przy tak dużych odchyleniach trudno obserwować ekran monitora. Dlatego niezbędne przechylenie głowy powinno być mniejsze od 30 stopni w każdym kierunku.

Uwzględniając teraz fakt, że odchylenie następuje w obie strony, mierzona wartość przyspieszenia będzie się zmieniać od -0,5g do +0,5g. Sumaryczna zmiana przyspieszenia o 1g skutkuje zmianą współczynnika wypełnienia sygnału wyjściowego o 12,5%. Przy okresie sygnału wyjściowego ustalonym na 10,4ms, moment zmiany stanu na wyjściu PWM przesunie się o 1,3ms.

Wiemy już, o ile zmieni się czas impulsu wyjściowego. Przyszła pora na określenie, ilu punktom na ekranie powinno to odpowiadać. Rozdzielczość standardowych kart graficznych wynosi obecnie od 640x480 do 1024x768 punktów. Myszka powinna mieć jednak nieco większą czułość. Już wyjaśniam dlaczego.

Przypuśćmy, że podchodzimy do komputera. Pierwsze co robimy, to szukamy kursora na ekranie. Okazuje się, że jest przy

prawej krawędzi ekranu. Przesuwamy myszkę w lewo. Gdy podkładka się skończy, podnosimy myszkę do góry, przenosimy w prawo, kładziemy na podkładce i znowu przesuwamy w lewo. Te czynności powtarzamy tak długo, aż kursor znajdzie się tam, gdzie powinien.

A teraz wyobraźmy sobie, że nie możemy odrywać myszki od podłoża i mamy ograniczone pole do jej przesuwania. Takie warunki dość dobrze ilustrują sytuację przy posługiwaniu się myszką nagłówną. Załóżmy, że kursor znowu jest po prawej stronie ekranu, a myszka jak na złość z lewej strony pola. Co robimy? Przesuwamy myszkę w prawo. Wkrótce kursor, nie mogąc wyjść dalej „oprze” się o prawą krawędź ekranu. Przesuwamy myszkę jeszcze trochę, aż dotrzemy do prawej krawędzi podkładki. Teraz wracamy myszką na środek pola. Taki sam ruch wykona kursor ustawiając się na środku ekranu. I o to nam chodziło.

Aby jednak coś takiego było możliwe, konieczny jest pewien zapas rozdzielczości. Wstępnie założyłem, że czułość myszki powinna wynosić około 1300 punktów przy pełnym zakresie odchylenia głowy.

Przeanalizujmy teraz, jaki to ma wpływ na konstrukcję myszki. Przebieg PWM najłatwiej zdekodować zliczając za pomocą wewnętrznych timerów mikrokontrolera czas trwania stanu niskiego i wysokiego. Ustaliliśmy już, że czas trwania tych stanów zmieni się o 1,3ms. Dla uzyskania założonej czułości wewnętrzny timer procesora powinien w tym czasie zliczyć przynajmniej 1300 impulsów. Wynika z tego, że częstotliwość taktująca timer powinna wynosić około 1MHz. W procesorach rodziny MCS-51, a początkowo chciałem zastosować AT89C2051, najwyższa częstotliwość taktująca wynosi 1/12 częstotliwości kwarcu. Oznacza to konieczność taktowania procesora przebiegiem o częstotliwości 12MHz. W takich warunkach „mały atmelek” pobiera około 10mA, a to już za dużo jak na układ, który ma być zasilany z portu szeregowego komputera.

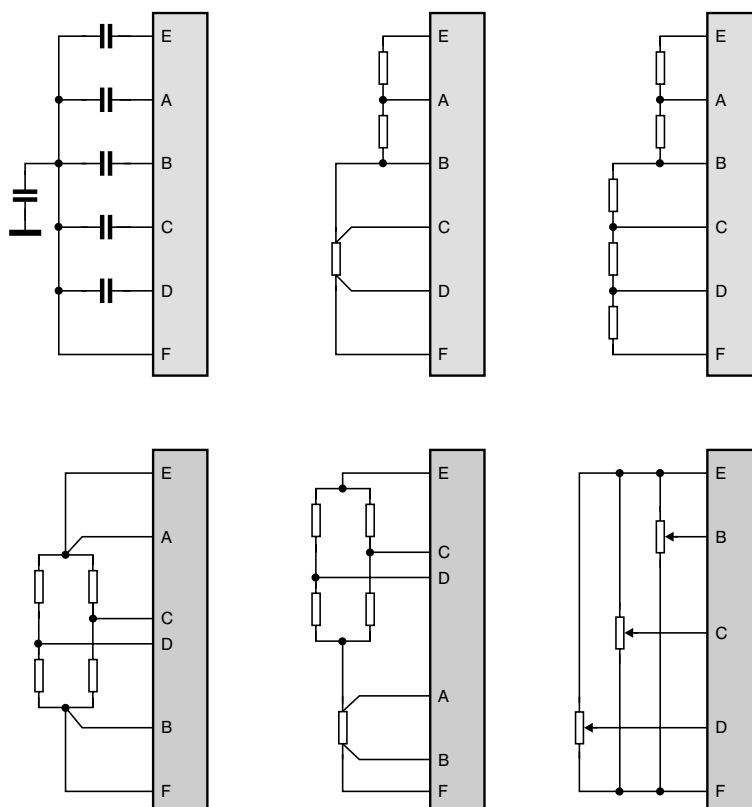
To zadecydowało o wyborze procesora AT90S2313 z rodziny AVR firmy Atmel. Przebieg taktujący wewnętrzne liczniki w tym procesorze może mieć częstotliwość równą częstotliwości rezonatora kwarcowego, czyli wystarczy taktowanie procesora kwarcem o częstotliwości 1MHz. Dlatego ta częstotliwość ostatecznie wzrosła do 3,58MHz, wyjaśnię później.

Gdy już miałem akcelerometry i procesor, zabrałem się za poszukiwanie przełączników ciśnieniowych. W tym momencie natrafiłem na nadspodziewanie wiele problemów. Z dmuchania w manometr ciśnieniomierza lekarskiego wiedziałem już, że potrzebuję przełączników o czułości rzędu 15mmHg (2kPa). Okazało się, że łatwiej znaleźć wyłączniki na zakres megapascali. W dodatku potrzebowałem przełączników różnicowych, w których zamknięcie zestyku następuje pod wpływem nadciśnienia (dmuchnięcie) lub podciśnienia (zassanie). Najbliższe spełnieniu tych warunków były niewielkie hydrostaty stosowane w sprzęcie AGD, np. w pralkach i ekspresach do kawy. Tu jednak trudno znaleźć jakiś stan-

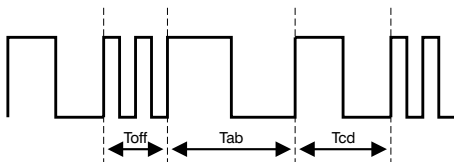
dardowy typ. W dodatku elementy te są zazwyczaj duże, bo przeznaczone do pracy przy znacznym obciążeniu i pod napięciem 220V.

Ostatecznie zdecydowałem się na zastosowanie czujnika ciśnienia MPX10DP firmy Motorola. Daje on wprawdzie sygnał wyjściowy na poziomie pojedynczych miliwoltów, jest nieskompensowany napięciowo i termicznie, za to jest najtańszy spośród całej rodziny czujników i nadaje się do mierzenia nad- i podciśnienia. Dodatkowo producent wypełnia wnętrze czujnika żel-silikonowym, dzięki czemu jest on w dużym stopniu odporny na wilgoć zawartą w powietrzu.

Koniecznym uzupełnieniem tego czujnika jest przetwornik analogowo-cyfrowy. W tej roli zastosowałem bardzo ciekawy układ UTI firmy Smartec. Nietypowe oznaczenie układu wynika chyba z faktu, że Smartec produkuje tak mały asortyment układów, że nie ma potrzeby ich numerować. Są wśród nich dość popularne czujniki temperatury typu SMT160-30 i mniej znane czujniki wilgotności względnej SMT-RH05.



Rys. 3. Sposoby podłączania różnych czujników do UTI.



Rys. 4. Przebieg na wyjściu układu UTI.

### Opis układu

Schemat elektryczny myszki znajduje się na rys. 2. Można na nim wyróżnić pięć bloków:

- czujnik przechyłu (U2),
- czujnik dmuchnięcia (S1) z przetwornikiem (U3),
- procesor sterujący (U1),
- interfejs we/wy (T1),
- zasilacz stabilizowany (U4).

Czujnik przechyłu zrealizowałem na układzie ADXL202. Jak wcześniej wspomniałem, układ ten zawiera dwa akcelerometry obrócone o kąt 90 stopni. Do poprawnej pracy układ potrzebuje dwóch kondensatorów filtrujących C8 i C9 oraz rezystora ustalającego R8.

Kondensatory filtrujące determinują czas odpowiedzi czujników przyspieszenia. C8 filtruje sygnał akcelerometru w jednej osi, podczas gdy C9 robi to samo w drugiej osi. Minimalna pojemność tych kondensatorów wynosi 1nF. Przy takiej pojemności akcelerometry mogą wykrywać drgania o częstotliwości 5kHz. W naszej aplikacji wystarczy czas reakcji rzędu dziesiątych części sekundy. Trudno bowiem sobie wyobrazić, aby ktoś mógł poruszać głową szybciej. Ograniczając pasmo przenoszenia czujników do 5Hz, zmniejszamy poziom szumów w sygnale wyjściowym. Przy wartości 1μF wartość szczytowa szumów nie powinna przekroczyć 8mg (g - przyspieszenie ziemskie). Ma to bardzo istotne znaczenie dla stabilności położenia kursora na ekranie. Przy zmianie mierzonego przyspieszenia w granicach 1g i rozdzielczości rzędu 1000 punktów, jeden piksel odpowiada 1mg. Poziom szumów na poziomie 8mg oznacza nieprzewidywalne ruchy kursora o 8 pikseli. Dalsze zwiększanie pojemności kondensatorów C8 i C9 nie powoduje już znaczącego zmniejszenia szumów. Lepsze efekty można uzyskać na drodze programowej, ale o tym pomówimy w części poświęconej oprogramowaniu.

Rezystor R8 ustala okres sygnału wyjściowego w obu kanałach. Przy wartości 1,3MΩ okres sygnału wyjściowego wynosi około 10,4ms. Wyjścia akcelerometrów 9 i 10 (U2) są połączone bezpośrednio z wejściami INTO (6) i INT1 (7) procesora (U1).

Drugi blok to czujnik ciśnienia S1 z przetwornikiem analogowo-cyfrowym U3. Konstrukcja sensora opiera się na klasycznym mostku rezystancyjnym o stopniu nierównoważenia zależnym od przyłożonego ciśnienia. Czujnik jest wyposażony w dwa króćce, z których pierwszy doprowadza medium do komory z jednej strony membrany czujnikowej, a drugi do komory z drugiej strony membrany.

Rezystancja mostka widziana od strony zasilania wynosi około 500Ω. Przy zasilaniu z +5V czujnik pobierałby 10mA, a to stanowczo za dużo. Konieczne było zatem znalezienie sposobu na obniżenie pobieranej mocy. Można to zrobić przez obniżenie napięcia zasilającego mostek. Wadą takiego rozwiązania jest zmniejszanie i tak już małego sygnału wyjściowego czujnika. Tu ujawniły się kolejne zalety zastosowanego przetwornika UTI. Ustawienie wejścia \PD (11, U3) w stan niski powoduje nie tylko uspienie przetwornika i zmniejszenie pobieranego prądu, ale również wyłączenie zasilania sensora! Usypianiem przetwornika steruje mikrokontroler za pomocą wyjścia PB3 (15, U1) Rezystor R7 wymusza niski stan na tej linii zaraz po pojawieniu się napięcia zasilającego i przez cały czas trwania impulsu zerującego procesor.

Czujnik S1 jest zasilany z układu UTI (U3) napięciem o przebiegu prostokątnym występującym na wyjściach E-F. Rzeczywista wartość napięcia zasilającego mostek jest mierzona na wejściach A-B. Takie rozwiązanie umożliwia prowadzenie pomiarów linią czteroprzewodową i znaczne oddalenie sensora od przetwornika. Właściwe napięcie nierównoważenia wchodzi na wejścia C-D.

Wyjście przetwornika jest podłączone do pinu PD6 (11, U1).

Nazwa UTI jest skrótem od *Universal Transducer Interface* i muszę przyznać, że nazwa nie jest na wyrost. Do tego przetwor-

nika można podłączać najróżniejsze czujniki: pojemnościowe, rezystory platynowe, termistory, mostki rezystancyjne lub po prostu potencjometry. Przykładowe sposoby podłączenia czujników przedstawiono na rys. 3. Układ posiada jedno wyjście, na którym pojawia się kilkufazowy przebieg umożliwiający kontrolerowi autokalibrację i pomiar. Dzięki temu rozdzielczość i liniowość pomiaru jest na poziomie 12 bitów.

W naszej myszce układ UTI pracuje w trybie pomiaru mostka rezystancyjnego o nierównoważeniu mniejszym niż ±4%. Czas pomiaru wynosi około 12ms. Przy takich ustawieniach na wyjściu pojawia się trójfazowy przebieg (rys. 4), w którym czas pierwszej fazy *Toff* umożliwia pomiar offsetu toru pomiarowego, czas drugiej fazy *Tab* określa wartość napięcia zasilającego mostek pomiarowy, a czas trzeciej fazy *Tcd* odpowiada napięciu wyjściowemu mostka. Znając te trzy czasy, można precyzyjnie obliczyć stopień nierównoważenia mostka.

Sercem myszki, i to szybko bijącym, jest mikrokontroler AT90S2313 taktowany sygnałem o częstotliwości 3,58MHz. Częstotliwość przebiegu zegarowego jest ustalana rezonatorem kwarcowym Q1 z towarzyszącymi kondensatorami C1 i C2. Ten typ procesora posiada 2KB pamięci programu, 128 bajtów pamięci RAM i tyle samo pamięci EEPROM.

Oprócz wspomnianych wcześniej sygnałów z akcelerometrów i przetwornika UTI, do procesora wchodzi jeszcze dwie linie sygnałowe. Zwarcie przewodów na złączu Z1 jest równoznaczne z naciśnięciem lewego klawisza myszki. Aby zasymulować naciśnięcie prawego klawisza, należy zewrzeć wyprowadzenia złącza Z6. Wejścia PB4 (16) i PB2 (14), którymi procesor (U1) czyta stan tych linii, są programowo ustawione jako wejścia z wewnętrznym podciągnięciem do plusa zasilania. Rezystory R1 i R2 zabezpieczają w pewnym stopniu te wejścia przed do końca przemyślanych eksperymentach.

Naciśnięcie wyłącznika W1 - „USTAW“ zwiera do masy wejście PD5 (9, U1). Przycisk ten służy do

zerowania czujnika ciśnienia. Kiedy i po co należy go nacisnąć, dowiemy się w części poświęconej uruchamianiu myszki.

Komunikację z komputerem zapewniają myszce dwie linie. Komputer, wystawiając na chwilę (około 100ms) jedynek logiczną (-12V) na wyjściu RTS (Z7), żąda od myszki nadania komunikatu identyfikującego. Sygnał ten przez rezystor R3 trafia na diody D4, D5 ograniczające jego amplitudę. Tak ukształtowany sygnał wchodzi na wejście zerujące procesora \RST (1, U1). Jeśli teraz komputer „zdejmie“ jedynek i na wyjściu pojawi się napięcie +12V (oznaczające zero), mikroprocesor rozpocznie wykonywanie zasytowanego programu, a tam jedną z pierwszych czynności jest wysłanie identyfikatora do komputera.

Transmisja w drugą stronę wychodzi z procesora nóżką TXD (3, U1) i przez dzielnik złożony z rezystorów R6 i R5 wchodzi na bazę tranzystora T1. Tranzystor ten pełni rolę konwertera poziomów. Gdy na wyjściu TXD procesora jest niski stan logiczny, tranzystor przewodzi i na jego kolektorze występuje napięcie bliskie napięciu zasilania +5V. Kolektor T1 jest połączony za pośrednictwem złącza Z2 z wejściem RD portu RS232C w komputerze, zatem komputer odczyta taki stan jako logiczne zero. Jeśli procesor U1 „wystawi“ na pinie TXD jedynek, tranzystor będzie zatkany i kolektor, a wraz z nim RD portu RS232C, będzie na potencjale -12V. To ujemne napięcie przychodzi na złącze Z3 z nieaktywnego w tym momencie wyjścia TD portu RS232C. Rezystor R4 ogranicza prąd pobierany z tego wyjścia, a dioda D1 przepuszcza tylko napięcie o ujemnej polaryzacji.

Ostatnim blokiem myszki jest stabilizator napięcia zasilającego. Źródłem zasilania dla myszki są dwie linie portu szeregowego: RTS (Z7) i DTR (Z4). Po uruchomieniu w komputerze programowego sterownika myszki, na obu liniach występuje niski stan logiczny, co odpowiada napięciu około +12V. Prąd z tych linii po przejściu przez diody D2 i D3 wchodzi na wejście stabilizatora U4. Wyjścia portu szeregowego charakteryzują się stosunkowo du-

żą rezystancją wyjściową, dlatego nie ma potrzeby stosowania rezystorów wyrównujących prąd pobierany z obu linii.

Podane wcześniej napięcia na wyjściach portu RS232C są wartościami nominalnymi. Już przy niewielkim obciążeniu napięcie spada do około  $\pm 10V$ . W komputerach, które sprawdziłem, przy obciążeniu rzędu 8mA napięcie nie spadało poniżej  $\pm 8V$ . Mniej więcej takiego napięcia potrzebuje na wejściu stabilizator U4 typu 78L05 do poprawnej pracy. Wprowadzić po drodze są jeszcze diody D2 i D3, ale dzięki zastosowaniu diod Schottky'ego spadek napięcia na nich jest pomijalnie mały (0,5V).

Znacznie korzystniej byłoby zastosować stabilizator typu „low-drop“, któremu do poprawnej pracy wystarczy, że napięcie zasilające będzie o kilkaset miliwoltów wyższe od napięcia wyjściowego. Pierwotnie planowałem zastosowanie w tym miejscu stabilizatora typu LM2931-5.0. Dość często można spotkać się z opinią, że układy te są mało stabilne i potrafią się wzbudzać. Skutkiem tego jest przegrzewanie elementu nawet przy teoretycznie niewielkiej rozdzielanej mocy. Zamierzałem rozprawić się z tą opinią, bo nigdy dotąd nie miałem z nimi problemów. Ten typ stabilizatora wymaga tylko nieco staranniejszego blokowania wyjścia. Powinno używać się do tego celu kondensatorów aluminiowych o małej wartości ESR lub kondensatorów tantalowych. Ku memu rozczarowaniu okazało się, że układ LM2931 jest dodatkowo uczulony na dużą rezystancję źródła zasilającego. Na nic się zdało blokowanie wyjścia i wejścia. Ostatecznie musiałem przeprosić stary pocziwy stabilizator 78L05.

Wejście stabilizatora oraz zasilanie wszystkich układów scalonych zablokowano kondensatorami C6, C5, C4 i C3.

Wyjście stabilizatora zostało zablokowane kondensatorem elektrolitycznym C7. Poprawia on odpowiedź stabilizatora na gwałtowne zmiany obciążenia. Musimy pamiętać, że myszka pobiera prąd impulsowo, a chwilowy pobór prądu może wynosić kilkanaście miliamperów. Pojemność tego kon-

densatora nie powinna być mniejsza od 100 $\mu F$ .

**Tomasz Gumny, AVT**  
tomasz.gumny@ep.com.pl

*„Dziękuję firmie ALFINE z Poznania za udostępnienie układów ADXL202 /Analog Devices/ i firmie UNIPROD-COMPONENTS z Gliwic za próbki układów UTI /Smartec/” - to zdanie napisałem posługując się opisaną w artykule myszką za pomocą programu wirtualnej klawiatury. Zajęło mi to 13 minut i 54 sekundy, co daje średnią szybkość pisania wynoszącą jeden znak na 5,5 sekundy.*

*Dziękujemy Panu Bogdanowi Janiakowi za pomoc w przygotowaniu zdjęcia na naszą kwietniową okładkę - Redakcja EP.*

*Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/pcb.html> oraz na płycie CD-EP04/2000 w katalogu PCB.*

## WYKAZ ELEMENTÓW

### Rezystory

R1, R2: 1k $\Omega$ /0,25W  
R3, R4: 10k $\Omega$ /0,25W  
R5, R6, R7: 100k $\Omega$ /0,25W  
R8: 1,3M $\Omega$ /0,25W

### Kondensatory

C1, C2: 22pF  
C3, C4, C5: 100nF/63V  
C6, C7: 100 $\mu F$ /16V  
C8, C9: 1 $\mu F$ /63V

### Półprzewodniki

D1, D2, D3, D4, D5: BAT85  
S1: MPX10DP  
T1: BC558  
U1: AT90S2313-10PC (zaprogramowany)  
U2: ADXL202JQC  
U3: UTI  
U4: 78L05

### Różne

Q1: rezonator kwarcowy 3,579545MHz  
Z1, Z6: ARK2 do druku  
Z2, Z3, Z4, Z5, Z7, Z9: kołki lutownicze  
podstawka pod układ scalony DIL20  
podstawka pod układ scalony DIL16  
mikroprzełącznik do druku  
Gniazdo D-SUB 9pin z obudową  
Kabel 5-żyłowy o długości 3m



# Myszka komputerowa dla osób niepełnosprawnych, część 2

## AVT-862



*W drugiej części artykułu omawiamy oprogramowanie mikrokontrolera „myszki” oraz jej montaż i uruchomienie.*

*Za miesiąc przedstawimy w EP program „wirtualnej klawiatury”, którego zastosowanie ułatwi korzystanie z komputera osobom niepełnosprawnym.*

### Oprogramowanie

Program dla mikroprocesora napisałem w języku C. Do kompilacji używałem produktu firmy ImageCraft ([www.imagecraft.com/software](http://www.imagecraft.com/software)), noszącego nazwę IC-CAVR. Z czystym sumieniem mogę polecić ten kompilator amatorom procesorów AVR. Prosta obsługa pozwala rozpocząć prace natychmiast po zainstalowaniu programu. Doskonale została rozwiązana diagnostyka błędów rozpoznawanych na etapie kompilacji i linkowania. Oprócz plików do programowania pamięci programu typu *.hex*, kompilator może generować pliki *.cof*, które są akceptowane przez symulator AVR-Studio. Dzięki temu wstępne uruchamianie można prowadzić na poziomie kodu źródłowego, co znacznie przyspiesza pracę i poprawia jej komfort. Należy do tego dodać dobrą jakość generowanego kodu i dostęp do postaci źródłowej bibliotek standardowych.

Program sterujący składa się z siedmiu części. Na początku procesor inicjalizuje stos i zmienne. Zaraz potem następuje odesła-

nie do komputera identyfikatora myszy. Pięć kolejnych bloków stanowi główną pętlę programu. Dwa pierwsze z nich są niemal identyczne i realizują odczyt wyjść akcelerometrów. W kolejnym, program interpretuje trójfazowy przebieg na wyjściu przetwornika UTI. Procesor musi też sprawdzić, czy nie zostały zwarte opcjonalne przełączniki, odpowiadające lewemu i prawemu klawiszowi myszki. Na końcu to wszystko jest upychane w jednym, trzypobytowym komunikacie, wysyłane do komputera i program wraca na początek pętli.

Na pierwszy rzut oka program wydaje się dość prosty. W rzeczywistości napisanie i uruchomienie zajęło mi tyle czasu, że 30-dniowa wersja demonstracyjna programu kompilatora przestała działać i zostałem niejako zmuszony do zakupu jego pełnej wersji.

O właściwą inicjalizację zmiennych i stosu musi zadbać sam kompilator, zatem program rozpocząłem od funkcji realizującej zgłoszenie komputerowi obecności myszki. W standardzie Microsoft Serial Mouse myszka zgłasza się

wysyłając literę „M“ (77 ASCII) po każdym ujemnym impulsie na linii RTS (Z7). Transmisja odbywa się z szybkością 1200 bodów. Nadawane słowo zawiera bit startu, siedem bitów danych i dwa bity stopu. Identyfikator ma zatem postać ciągu bitów: 0\_1110111\_11. Procesor AT90S2313 jest wyposażony w port szeregowy. Dzięki temu cała procedura identyfikacji sprowadziła się do ustawienia parametrów transmisji, załadowania bufora i czekania na sygnał końca nadawania.

Kolejnym zadaniem programu jest odczyt sygnałów z akcelerometrów. Wyjścia obu akcelerometrów są podłączone do wejść przerwania zewnętrznych INT0 (6, U1) i INT1 (7, U1). Program najpierw ustawia wejście INT0 jako czułe na zbocze opadające i wykonywanie programu zostaje wstrzymane instrukcją SLEEP. Gdy na wejściu pojawi się takie zbocze, procesor budzi się i przechodzi do obsługi przerwania. Ta i wszystkie pozostałe procedury obsługi przerwania są puste. Dzięki temu procesor szybko wraca do programu głównego. Teraz jest uruchamiany wewnętrzny, 16-bitowy Timer1. Wejście INT0 uczulane jest na zbocze narastające i procesor jest ponownie usypiany. Zbocze kończące impuls z akcelerometru budzi procesor, który zatrzymuje Timer1 i zapamiętuje jego zawartość. W ten sposób mierzony jest czas trwania ujemnego impulsu na wyjściu oznaczonym przez producenta jako OUTX (10, U2). W naszym układzie jest to akurat wyjście akcelerometru czułego na pochylenie głowy do przodu i do tyłu, czyli w osi Y.

Fragment programu realizujący ten pomiar można prześledzić

na list. 1. Zainteresowanym Czytelnikom chciałbym zwrócić uwagę na sposób wstawiania instrukcji asemblerowych, bo jest to kolejna zaleta kompilatora IC-CAVR.

Taki sam cykl powtarza się dla akcelerometru OUTY (9, U2), podłączonego do wejścia INT1. Różnica polega tylko na tym, że mierzony jest czas trwania impulsu dodatniego.

Tutaj należy się kilka słów wyjaśnienia. Zasadniczo, aby móc obliczyć wartość przyspieszenia, konieczne jest mierzenie dwóch parametrów impulsów: czasu trwania impulsu i okresu powtarzania całego przebiegu. Dopiero iloraz tych czasów wskazuje dokładny wynik pomiaru. W naszym przypadku taka dokładność nie jest konieczna. Z dobrym przybliżeniem możemy założyć, że okres przebiegu na wyjściu akcelerometrów jest stały. Taki stały czynnik można pominąć, gdyż nie interesuje nas bezwzględna wartość przyspieszenia, a tylko zmiana tej wartości. Zależnie od tego, czy będziemy mierzyć impulsy ujemne, czy dodatnie, wzrost przyspieszenia będzie powodował zwiększenie lub zmniejszanie współrzędnych kursora na ekranie.

Aby zmniejszyć błędy wynikające z bramkowania, zawartość licznika jest dzielona przez cztery. Dla zachowania rozdzielczości konieczne było podwyższenie częstotliwości taktującej. Wybrałem kwarc o „telewizyjnej“ częstotliwości 3,58MHz.

Wcześniej sygnalizowałem problem szumów na wyjściach akcelerometrów. Nie chodzi o szumy w tradycyjnym rozumieniu tego słowa. „Szumienie“ akcelerometrów objawia się losową zmianą długości im-

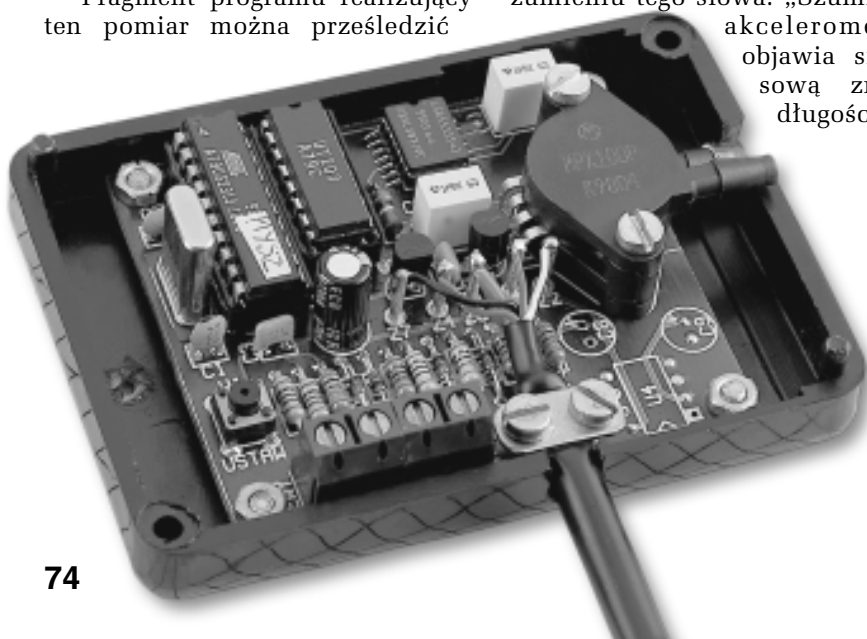
List.1. Podprogram realizujący pomiar przechylenia w osi Y.

```

/** Pomiar przechylenia w osi Y ***/
GIMSK=0x00; /* INT0: wyłącz */
MCUCR=0x22; /* INT0: 1->0 */
GIMSK=0x40; /* INT0: włącz */
asm("SLEEP"); /* czekaj na 1->0 */
asm("NOP");
GIMSK=0x00; /* INT0: wyłącz */
TCCR1B=0x00; /* TIMER1: stop */
TCNT1=0x0000; /* TIMER1: zeruj */
TCCR1B=0x01; /* TIMER1: CK / 1 */
MCUCR=0x23; /* INT0: 0->1 */
GIMSK=0x40; /* INT0: włącz */
asm("SLEEP"); /* czekaj na 0->1 */
asm("NOP");
GIMSK=0x00; /* INT0: wyłącz */
TCCR1B=0x00; /* TIMER1: stop */
probkiY[lprob]=TCNT1; /* nowy czas */
sumaY=0; /* sumuj próbki */
for(i=0;i<LPROB;i++)
    sumaY+=probkiY[i];
deltaY=(sumaY-psumaY)/DZIEL;
if(deltaY>127)
    deltaY=127; /* tyle można wysłać */
if(deltaY<-128)
    deltaY=-128;
psumaY=sumaY;
    
```

pulsów wyjściowych. Program zmniejsza efekty tego zjawiska przez uśrednianie wyników z ośmiu ostatnich pomiarów. Powoduje to wprawdzie dodatkowe opóźnienie reakcji myszki na ruchy głową, ale wyszedłem z założenia, że ważniejsza jest możliwość precyzyjnego ustawienia kursora.

Nieco bardziej rozbudowany jest fragment programu odczytujący przebieg wyjściowy z przetwornika UT1. Przebieg ten składa się z trzech faz, różniących się czasem trwania. Kształt przebiegu ilustruje rys. 4. Pierwsza faza, oznaczona jako *Toff*, wskazuje offset toru pomiarowego. Dla wyróżnienia składa się ona z dwóch identycznych okresów. Druga faza - *Tab* - służy do pomiaru napięcia zasilającego czujnik. Ostatnia, trzecia faza, określa napięcie niezrównoważenia mostka pomiarowego i jest oznaczona jako *Tcd*. Ze względu na ściśle czasowy charakter przebiegu, tutaj również wykorzystałem timer T1 z tym, że licznik zlicza impulsy bez zatrzymywania. Sygnał wyjściowy z przetwornika UT1 jest podawany do procesora na wyprowadzenie o nazwie ICP - Input Capture Pin (11, U1). Przy odpowiednim skonfigurowaniu tego wejścia, każde pojawiające się na nim narastające zbocze powoduje wygenerowanie przerwania i równoczesne przepisanie aktualnej zawartości licznika T1 do specjalnego rejestru ICR. Poprzednia zawartość licznika jest odejmowana od aktualnej i ta różnica jest cyklicznie zapisywana do 4-elementowej tablicy. Po każdym przerwaniu procesor sprawdza, czy pierwsze dwa elementy





spełniają warunki nałożone na fazę *Toff*. Jeśli tak, procesor może wyliczyć wartość ciśnienia ze wzoru:

$$P = 1/32 * (Tcd - Toff) / (Tab - Toff)$$

Podobnie jak w przypadku akcelerometrów, tutaj również nie jest nam potrzebna bezwzględna wartość ciśnienia. Wystarczy, jeśli będziemy sprawdzać, czy naciśnięcie (lub podciśnienie) nie przekracza pewnej wartości. W tym celu procesor musi znać spoczynkową wartość (*Tcd-Toff*). Taki wzorcowy pomiar jest dokonywany jeden raz, a wynik zostaje zapisany w pamięci nieulotnej EEPROM. Sposób przeprowadzenia kalibracji zostanie opisany nieco później. Przy kolejnych pomiarach aktualna wartość (*Tcd-Toff*) jest porównywana z wartością odniesienia. Jeśli obliczone w ten sposób naciśnięcie przekracza wartość minimalną, jest to interpretowane jako naciśnięcie lewego przycisku myszki. Jeśli natomiast w czujniku pojawi się podciśnienie o odpowiednio dużej wartości, program przyjmie, że naciśnięto przycisk prawy.

Niezależnie od pomiaru ciśnienia, procesor sprawdza stan dwóch pinów PB2 (14, U1) i PB4 (16, U1). Wejścia te mają włączony wewnętrzny podciąganie do plusa zasilania i w spoczynku występuje na nich poziom wysoki. Jeśli teraz zewrzymy wyprowadzenia złącza Z1, na wejściu PB4 pojawi się poziom niski i procesor odczyta to jako naciśnięcie lewego przycisku. Analogicznie zwarcie złącza Z6 zostanie potraktowane jako naciśnięcie prawego klawisza myszki.

Programowa obsługa przycisku USTAW (W1) jest realizowana w dwóch blokach programu. Pierwszy raz jego stan jest sprawdzany na początku programu, na etapie inicjalizacji. Jeśli procesor stwierdzi, że styki tego przycisku są zwarte, blokuje odczytywanie przetwornika UTI. W takim przypadku jedynym sposobem naciśnięcia klawiszy myszki jest zwiernie złącza Z1 i Z6.

Podczas normalnej pracy stan mikroprzełącznika W1 jest sprawdzany po każdym pomiarze ciśnienia. Zwarcie styków w takim momencie powoduje zatrzymanie programu i zapis aktualnego ci-

**Tab.2. Struktura komunikatu w standardzie MS-Mouse.**

| bajt/bit | 7 | 6 | 5  | 4  | 3  | 2  | 1  | 0  |
|----------|---|---|----|----|----|----|----|----|
| 1        | 1 | 1 | LB | RB | Y7 | Y6 | X7 | X6 |
| 2        | 1 | 0 | X5 | X4 | X3 | X2 | X1 | X0 |
| 3        | 1 | 0 | Y5 | Y4 | Y3 | Y2 | Y1 | Y0 |

nienia do pamięci nieulotnej. Tak zapamiętana wartość jest wartością odniesienia przy interpretowaniu następnym pomiarów. Po zwolnieniu przycisku program kontynuuje działanie z nową zawartością pamięci EEPROM.

W ostatnim kroku program musi złożyć rezultaty pomiaru przechylić w obu osiach z wynikami pomiaru ciśnienia. Potem trzeba dodać do tego stan na złączach przycisków dodatkowych i skompletowany w ten sposób komunikat można wysłać do komputera. W standardzie MS-Mouse myszka wysłała do komputera trzy bajty. Pierwszy bajt zawiera informacje o stanie klawiszy i dwa najstarsze bity przesunięcia w poziomie i pionie. W drugim bajcie znajduje się sześć młodszych bitów przesunięcia w kierunku X, a w trzecim to samo dla osi Y. Obrazowo pokazano to w **tab. 2**. Liczby reprezentowane są w kodzie uzupełnieniowym do dwóch. Oznacza to, że między jednym a drugim komunikatem kursor może się przemieścić maksymalnie o +127 lub -128 punktów. Jeśli przemieszczenie będzie większe, program ograniczy je do dopuszczalnego zakresu. Widać to dokładnie na list. 1. Pozycje oznaczone w tab. 2. symbolami LB (Left Button) i RB (Right Button) niosą informację o stanie klawiszy. Jedyńka odpowiada naciśnięciu klawiszowi.

Podczas eksploatacji modelu myszki miałem duże kłopoty z uzyskaniem podwójnego kliknięcia. Początkowo sądziłem, że zbyt duży jest odstęp między jednym a drugim dmuchnięciem. Niestety, ten sam efekt występował przy użyciu zwykłego przycisku podłączonego do złącza Z1. Szybko okazało się, że aby komputer odczytał dwa dmuchnięcia jako jedno podwójne kliknięcie, kursor musi pozostawać przez cały czas idealnie nieruchomy. Mimo że dużo wysiłku włożyłem w wyeliminowanie przypadkowych skoków kursora spowodowanych szu-

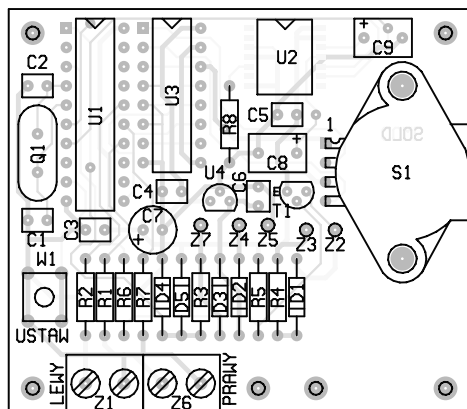
mem na wyjściu akcelerometrów, to ruchy o 1..2 punkty są nieuniknione. Problem podwójnego kliknięcia rozwiązałem w ten sposób, że po każdej zmianie stanu klawiszy ruchy kursora są blokowane na około 0,5 sekundy. Podnosi to znacząco komfort pracy, tylko nieznacznie ją spowalniając.

## Montaż

Wszystkie podzespoły myszki montujemy na dwustronnej płytce drukowanej, której układ ścieżek można znaleźć na wkładce wewnątrz numeru, płyty CD-ROM dołączonej do tego numeru EP lub pod adresem [www.ep.com.pl/pcb.htm](http://www.ep.com.pl/pcb.htm). Rozmieszczenie elementów ilustruje **rys. 5**. Na rysunku nie zaznaczono biegunowości kondensatora elektrolitycznego C6. Dodatni biegun tego kondensatora znajduje się od strony złącza Z5.

Jako złącza: Z2, Z3, Z4, Z5 i Z7 przewidziałem zastosowanie kołków lutowniczych. Nie jest to konieczne, ale ich obecność może później ułatwić ewentualną wymianę uszkodzonego kabla połączeniowego. Montaż proponuję zacząć od ostrożnego wbicia tych pięciu szpilek w płytkę drukowaną.

Następnie montujemy elementy w kolejności od najniższych do najwyższych. Proponuję nie montować na razie rezystora R8, podstawkę pod U3, kondensatora C9 i czujnika S1, gdyż będą one przeszkadzać przy lutowaniu sensora U2. Element ten jest w obudowie do montażu powierzchniowego, dlatego trzeba poświęcić mu trochę więcej uwagi. Oczywiście najlepsza byłaby lutownica na gorące powietrze i pasta lutownicza. W amatorskich warunkach wystarczająco dobre rezultaty można osiągnąć stosując cynę w postaci wielordzeniowego drutu o średnicy 1mm lub mniejszej i lutownicę wyposażoną w ostro zakończony grot. Zwykle zdajemy sobie sprawę, że w pracy z układami scalonymi należy stosować środki ostrożności zapobiegające



Rys.5. Rozmieszczenie elementów na płytce drukowanej.

powstawaniu ładunków elektryczności statycznej. W praktyce różnie z tym bywa i zwykle nic wielkiego się nie dzieje. Chciałbym ostrzec, że ADXL202 są nieco bardziej czułe na elektryczność statyczną niż inne układy. Dlatego nie od rzeczy będzie uziemiona lutownica i opaska uziemiająca na rękę. Lutujemy najpierw jeden pin i dokładnie pozycjonujemy układ. Teraz lutujemy przeciwległe wyprowadzenie i ponownie sprawdzamy, czy końcówki leżą dokładnie na pocynowanych plackach miedzi. Jeśli tak jest, lutujemy pozostałe piny używając przy tym jak najmniejszej ilości cyny.

W czujniku ciśnienia S1 ostrożnie zaginamy wyprowadzenia w odległości około jednego milimetra od obudowy. Końcówka numer jeden jest półkuliście nacięta, dlatego nie powinno być problemów z jej identyfikacją. Przykręcamy obudowę czujnika dwoma śrubkami M3 i dopiero potem lutujemy wyprowadzenia do płytki.

Na koniec montujemy pozostałe brakujące elementy pamiętając, że pod procesor U1 i przetwornik U3 należy zamontować podstawkę. Złącza śrubowe Z1 i Z6 proponuję wlotować w taki sposób, aby kabel wchodził do nich od środka płytki. Dzięki temu, jeśli zdecydujemy się na dołączenie do myszki dodatkowych przycisków, przewody do nich będzie można docisnąć do obwodu drukowanego razem z kablem RS232C. Posłuży do tego blaszka i dwie śrubki M3. Odpowiednie otwory do tego celu znajdują się na płytce. Takie rozwiązanie skutecz-

nie zabezpieczy przewody przed wyrwaniem, a o to w czasie eksploatacji naprawdę nietrudno.

## Uruchomienie i kalibracja

Uruchomienie jak zawsze rozpoczynamy od sprawdzenia poprawności montażu. Jeśli nie ma zwarcień między polami lutowniczymi i wszystkie elementy wydają się być obsadzone poprawnie, wyciągamy z podstawki procesor U1 i przetwornik U3. Do tak przygotowanego układu możemy podłączyć zasilanie. Najlepiej do tego celu wykorzystać regulowany zasilacz. Plus zasilacza podłączamy do złącza Z4, a masę do Z5. Włączamy zasilacz i stopniowo zwiększamy napięcie wyjściowe, cały czas sprawdzając napięcie na wyjściu stabilizatora U4. Po przekroczeniu 8V na zasilaczu, napięcie wyjściowe U4 powinno ustabilizować się na poziomie 5V. W tych warunkach pobierany prąd nie powinien przekraczać 2mA. Jeśli wyniki pomiarów są zgodne z oczekiwaniami, wyłączamy zasilanie i wkładamy w podstawki procesor i przetwornik.

Przy wyłączonym komputerze podłączamy naszą myszkę do portu RS232C i włączamy komputer. Pomimo że myszka leży nieruchomo, kursor może wykonywać niewielkie ruchy na ekranie. Jest to normalny objaw i świadczy o poprawnym działaniu. Przy pierwszym włączeniu myszka będzie się zachowywać tak, jakby cały czas miała naciśnięty któryś klawisz. Dlatego nie ruszając jej z miejsca, powinniśmy czym prędzej dokonać kalibracji czujnika ciśnienia. W tym celu upewniamy się, że nikt nie dmucha w rurkę czujnika i naciskamy na chwilę przycisk USTAW (W1). Na czas wciśnięcia przycisku kursor powinien znieruchomieć. Wynik kalibracji jest zapisywany w pamięci nieulotnej, dlatego wystarczy wykonać ją tylko raz. Od tego momentu myszka jest gotowa do pracy.

## Konstrukcja mechaniczna

Nie da się ukryć, że myszka w postaci płytki drukowanej z zamontowanymi elementami jest właściwie bezużyteczna. Dlatego

chciałbym przedstawić kilka rozwiązań mechanicznych, które zastosowałem w modelu. Przede wszystkim myszkę należy zamknąć w niewielkiej obudowie z tworzywa sztucznego. Z obudowy powinny wystawać oba króćce czujnika ciśnienia. Standardowo rurkę podłączamy do tego, który znajduje się bliżej płytki drukowanej.

Do stabilnego zamocowania myszki na głowie wykorzystałem zwykłą czapkę „bejsbolówkę“ założoną daszkiem do tyłu. Na górze czapki zostały przyszyte rzepy (dziękuję Aniu!). Drugą część rzepów przykleiłem do spodu obudowy już samodzielnie. W daszku czapki wykonałem kilka otworów. Dwa z nich służą do zamocowania kabla do komputera. Dzięki temu ten dość długi przewód nie ściąga myszki do tyłu przy każdym ruchu. Mniejsza jest też szansa, że myszka wylądnie na podłodze, gdy ktoś zahaczy o kabel.

Pozostałe otwory służą do zamocowania pałaka z dość twardego drutu. Pałak przechodzi koło ucha i kończy się na wysokości ust spłaszczonym oczkiem. W to oczko wciśnięty jest ustnik, który wykonałem ze zbiorniczka „aparatu do iniekcji“, czyli popularnej kroplówki. Podgrzany w gorącej wodzie zbiorniczek spłaszczyłem w połowie wysokości. Po odcięciu górnej pokrywy powstał całkiem zgrabny ustnik. Z drugiej strony zbiorniczka wychodzi wężyk idealnie pasujący do czujnika ciśnienia.

Czujnik ciśnienia jest w dużym stopniu odporny na działanie wilgoci. Mimo to ustnik i rurkę po umyciu należy każdorazowo przedmuchać i dokładnie osuszyć.

## Możliwości adaptacji

Konstrukcja i oprogramowanie myszki daje dość duże możliwości adaptacji do indywidualnych potrzeb.

Zacznijmy od czułości. Przez zmianę wartości rezystora R8 możemy wpływać na okres sygnału wyjściowego z akcelerometrów. Przy minimalnej dopuszczalnej rezystancji, wynoszącej 125kΩ, zmniejszamy czułość dziesięciokrotnie. Jakkolwiek największą wartością zalecaną dla aplikacji

wykorzystujących wyjścia PWM jest  $1,25M\Omega$ , to podłączając rezystor  $2M\Omega$  można uzyskać całkiem stabilne zwiększenie czułości.

W celu przystosowania myszki dla osób, u których występuje bezwiedne drżenie głowy, można zwiększyć pojemności kondensatorów C8, C9. Efektem tego będzie wolniejsza reakcja myszki na ruch głowy, przy czym czułość nie ulegnie zmianie. Stosując kondensatory elektrolityczne należy zwrócić uwagę na biegunowość. Odpowiednie symbole znajdują się na płycie drukowanej.

W modelu myszki obsługa klawiszy polega na dmuchaniu lub zasysaniu powietrza z rurki podłączonej do czujnika ciśnienia. Dmuchięcie odpowiada lewemu, a zassanie prawemu klawiszowi. Działanie możemy w prosty sposób odwrócić podłączając rurkę do drugiego króćca czujnika.

Jeszcze innym rozwiązaniem jest podłączenie rurek do obu

króćców. W takim przypadku dmuchnięcie w jedną rurkę odpowiada wciśnięciu lewego klawisza, a dmuchnięcie w drugą jest tożsame z naciśnięciem prawego klawisza. Dla osób z częściowo sprawnymi rękami można rurki przedłużyć i zakończyć gumowymi gruszkami. W ten sposób po ponownym skalibrowaniu czujnika ciśnienia przyciskiem USTAW, otrzymamy wygodne w obsłudze wyłączniki pneumatyczne.

Jeśli komuś wystarczą wyłączniki elektryczne, nie ma potrzeby stosowania „instalacji pneumatycznej“. Można wówczas znacząco obniżyć koszt myszki rezygnując z montowania czujnika S1 i przetwornika U3. W takim przypadku trzeba kawałkiem przewodu zewrzeć na stałe wyprowadzenia mikroprzełącznika W1. W momencie inicjalizacji procesor sprawdza stan tego przycisku i jeśli jest on zwarty, pomija w dalszej

pracy procedury odczytu czujnika ciśnienia.

**Tomasz Gumny, AVT**  
**tomasz.gumny@ep.com.pl**

*"Dziękuję firmie ALFINE z Poznania za udostępnienie układów ADXL202 /Analog Devices/ i firmie UNIPROD-COMPONENTS z Gliwic za próbki układów UTI /Smartec/" - to zdanie napisałem posługując się opisaną w artykule myszką za pomocą programu wirtualnej klawiatury. Zajęło mi to 13 minut i 54 sekundy, co daje średnią szybkość pisania wynoszącą jeden znak na 5,5 sekundy.*

*Dziękujemy Panu Bogdanowi Janiakowi za pomoc w przygotowaniu zdjęcia na naszą kwietniową okładkę - Redakcja EP.*

*Wzory płytek drukowanych w formacie PDF są dostępne w Internecie pod adresem: <http://www.ep.com.pl/pcb.html> oraz na płycie CD-EP05/2000 w katalogu PCB.*

# Myszka komputerowa dla osób niepełnosprawnych, część 3

## Wirtualna klawiatura

### AVT-862

*W dwóch poprzednich numerach EP opisaliśmy sterownik zastępujący komputerową mysz dla osób z niesprawnymi rękami.*

*Sterownik ten jest przystosowany do współpracy z graficznymi środowiskami użytkownika, przede wszystkim z Windows 95/98 i 2000.*

*W ostatniej części artykułu przedstawiamy program wirtualnej klawiatury, opracowany specjalnie z myślą o myszce dla niepełnosprawnych.*

W środowisku Windows myszka jest bardzo intensywnie wykorzystywana do kontroli pracy zarówno samego systemu, jak i uruchomionych w nim aplikacji. Służy do tego cały szereg wbudowanych mechanizmów:

- menu startowe umożliwiające uruchamianie programów,
- menu kontekstowe otwierane prawym przyciskiem myszy - pozwala na wykonanie wybranych czynności dotyczących obiektu, na którym kliknęliśmy,
- pasek zadań - realizuje szybkie przełączanie pomiędzy uruchomionymi programami,
- tradycyjne menu rozwijalne pozwala na szczegółową obsługę aplikacji,
- mechanizm „Przeciągnij i upuść” pozwala na łatwą manipulację plikami i innymi obiektami.

#### Ogólny opis programu

Program jest przewidziany dla jednego z najbardziej obecnie popularnych systemów operacyjnych: Microsoft Windows 95/98. Nie będzie działać w starszych, 16-bitowych „okienkach”.

Początkowo zamierzano wiernie odtworzyć na ekranie sprzętową klawiaturę (tak jak w popularnych programach do nauki szybkiego pisania). Jednak po pierwszych próbach oraz zebraniu wstępnych opinii i życzeń osób zainteresowanych okazało się, że koncepcję tę trzeba w dużym zakresie zmodyfikować. Po pierwsze: klawiatura tradycyjna zakłada używanie obu rąk. Nawet jeśli mamy mało wprawy i piszemy przysłowiowym „jednym palcem”, to i tak wciąż używamy klawiszy systemowych (*Shift*, *Ctrl*, *Alt*) dla uzyskania dodatkowego efektu (dużej litery, znaku polskiego, zaznaczenia fragmentu, przeskoku o cały wyraz itp.). Natomiast na ekranie mamy

w tym samym czasie możliwość „wciśnięcia” tylko jednego klawisza. Po drugie, dobrze jest wyposażać klawiaturę ekranową w dodatkowe możliwości, których sprzętowa nie posiada (np. dodatkowe schowki). Po uwzględnieniu powyższych założeń powstał wzór klawiatury, graficznie przypominający nieco panel foliowy (rys. 1), z rozmieszczeniem klawiszy uwzględniającym:

- oddzielne klawisze dla polskich znaków (zwykle wprowadzanych z użyciem *Alt*),
- wykorzystanie zwiększonych możliwości każdego klawisza (czyli oddzielna funkcja dla kliknięcia lewym i prawym przyciskiem myszy, np. dla liter jest to wpisanie małej lub dużej litery),
- ukrycie klawiszy rzadko używanych (klawisze funkcyjne są pokazywane tylko na życzenie),
- wprowadzenie nowych klawiszy nawigacyjnych (przesuwanie o wyraz - w typowych edytorach realizowane za pomocą *Ctrl*),
- zaznaczanie fragmentów tekstu za pomocą pojedynczych klawiszy (zazwyczaj jest to realizowane za pomocą *Shift*),
- wprowadzenie czterech, obsługiwanych bezpośrednio z klawiatury, dodatkowych schowków-zakładek do przechowywania fragmentów tekstu.

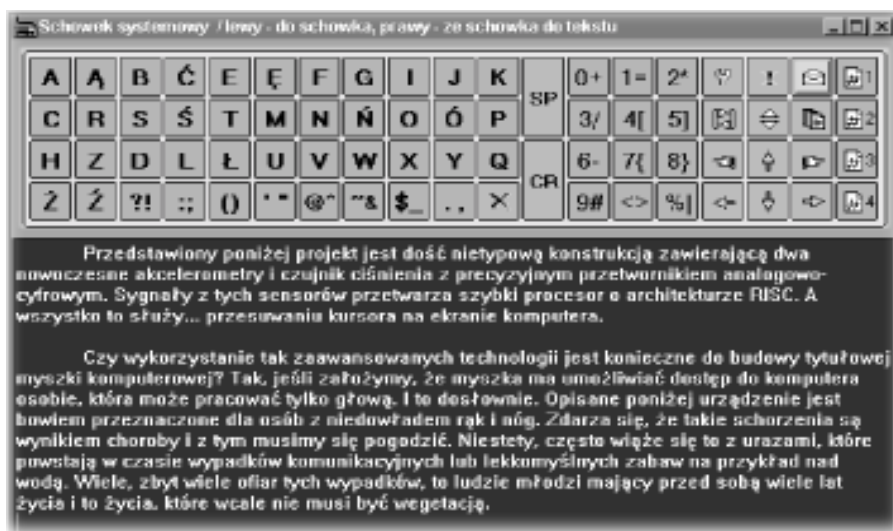
Poniżej pola klawiszy umieszczone jest okienko edycyjne, w którym wprowadzamy i opracowujemy tekst (rys. 1). Całe pokazane okno programu ma wymiary 640x480 pikseli, co umożliwia swobodną pracę w podstawowej rozdzielczości VGA.

#### Obsługa klawiatury za pomocą myszy

Posługiwanie się klawiaturą polega na wybraniu ruchem myszy odpowiedniego klawisza, a następn-







Rys. 1. Okno programu wirtualnej klawiatury.

nie kliknięcie lewym lub prawym przyciskiem myszy (w pełnej wersji myszki nagłownej jest to dmuchnięcie lub zassanie powietrza z ustnika). Wybrany klawisz jest zaznaczony jako szary przycisk na tle kolorowego pola. Wybieranie może się odbywać w dowolny sposób. Dla ułatwienia przemieszczania się po klawiaturze wprowadzono wspomagający mechanizm wybierania. Polega on na wygaszeniu kursora myszki i samoczynnym zaznaczeniu kolejnych klawiszy zgodnie z kierunkiem ruchu myszy. Szybkość zmiany zaznaczeń może być regulowana w celu bardziej precyzyjnego naprowadzenia. Niektórym z testujących program bardzo się to wspomaganie podobało, natomiast inni woleli wybierać klawisze tradycyjnie - celując kursorem. Dlatego jest to opcja przełączana - użytkownik wybiera sposób najbardziej pasujący do jego indywidualnych gustów i przyzwyczajajeń.

Przeznaczenie wybranego klawisza jest opisane w belce tytułowej okna. Po nabraniu wprawy w używaniu klawiatury pomoc ta może nieco nużyć - więc również jest możliwość jej wyłączenia.

Dodatkowym ułatwieniem w nawigacji po klawiaturze jest opcja ograniczania ruchu kursora do pola klawiszy (można ją włączyć i wyłączać na bieżąco prawym przyciskiem klawisza *Enter*). Eliminuje to przypadkowe „uciekanie” kursora poza klawiaturę i ułatwia okresowe wycentrowanie myszki.

Jak zaznaczono wcześniej, każdy klawisz posiada dwie funkcje - odpowiadające lewemu i prawe-

mu przyciskowi myszy. Oprócz tego działanie klawiszy ulega zmianie po przełączeniu się na tryb funkcyjny.

### Funkcje dodatkowe

Klawisze funkcyjne - jako rzadko używane - są normalnie ukryte. Można je wyświetlić (na miejscu numerycznej części klawiatury) używając lewego przycisku klawisza *Funkcje/Konfiguracja* (rys. 2). Klawisze funkcyjne mają inne znaczenie niż zazwyczaj. Nie służą do sterowania programami (nie jest przewidziane przekazywanie do innych uruchomionych aplikacji komunikatów związanych z naciśnięciem F1..F12), ale do dodatkowego sterowania pracą klawiatury. W tej wersji programu (2.8) wykorzystane są tylko niektóre funkcje:

- F1 - wyświetla plik pomocy,
- F10 - umożliwi samodzielną konfigurację układu liter na klawiaturze,
- F12 - przełącza klawiaturę w tryb pracy z zewnętrznymi aplikacjami.

### Tryby pracy klawiatury

Podstawowym, przewidzianym zadaniem klawiatury jest pisanie we własnym okienku edycyjnym (tak jak to widać na rys. 1). Wszystkie akcje klawiszy znakowych i nawigacyjnych dotyczą wtedy tego okienka. Przygotowane fragmenty tekstu przenosimy do docelowych aplikacji (edytor, program pocztowy, itp.) za pośrednictwem schowka systemowego (do obsługi którego służy oddzielny klawisz). W podobny sposób można ulokować tekst w wielu innych polach edycyjnych, o ile

wyposażone są one w menu kontekstowe z komendą *Wklej (Paste)*.

Jednakże w niektórych przypadkach schowek systemowy nie wystarcza. Tak dzieje się na przykład przy wpisywaniu haseł, czasem przy wypełnianiu formularzy itp. Dlatego też klawiatura została wyposażona w możliwość bezpośredniej współpracy z zewnętrzną aplikacją. Przełączenia trybów pracy dokonujemy (jak wspomniano wyżej) za pomocą klawisza funkcyjnego F12. Wtedy własne okienko edycyjne zostaje ukryte, natomiast wpisywane znaki są wysyłane bezpośrednio do ostatnio aktywnego programu. W praktyce polega to na ustawieniu myszą kursora tekstowego w potrzebnym miejscu.

Następnie - jeśli dodatkowy program nie zasłania okienka klawiatury - możemy od razu wpisywać tekst (rys. 3). Jeśli klawiatura chowa się za aktywnym oknem - przenosimy ją na wierzch za pomocą systemowego paska zadań i wtedy piszemy (tekst dotrze do programu, nawet jeśli klawiatura zasłoni miejsce wpisywania, ale oczywiście ze względów praktycznych lepiej jest tak ustawić okienka, żeby wszystko widzieć - rys. 4).

Należy zaznaczyć, że w edycji dłuższego tekstu praca w tym trybie jest dużo mniej wygodna. Nie jest widoczny kursor tekstowy w docelowym programie i nie wszystkie klawisze działają (akcje większości klawiszy polegają na wysłaniu odpowiednich komunikatów Windows, ale niektóre operują bezpośrednio na funkcjach wbudowanego okienka edycji - te drugie rzecz jasna nie mają żadnego wpływu na zewnętrzny program). Zalecane jest wtedy używanie trybu podstawowego.

### Konfiguracja programu

Każdy z użytkowników zazwyczaj ma swoje przyzwyczajenia dotyczące ustawień różnych parametrów programów (kolory, dźwięki, szybkość reakcji itd.). Dobrze jest, jeśli program pozwala



Rys. 2. Włączenie trybu funkcji dodatkowych.





Rys. 3. Współpraca z zewnętrznym oknem przy klawiaturze widocznej.

na samodzielny wybór takich ustawień. Klawiatura jest wyposażona w dodatkowe okno dialogowe konfiguracji, przedstawione na **rys. 5**, które otwieramy prawym przyciskiem na klawiszu *Funkcje/Konfiguracja*.

Szereg opcji dotyczy wybierania klawiszy:

- ✓ Wybieranie klawiszy kursorem - przełącza (jak wspomniano wcześniej) pomiędzy tradycyjnym naprowadzaniem kursora na klawisz a samoczynnym przewijaniem klawiszy zgodnie z ruchem myszki.
- ✓ Szybkość przewijania klawiszy w pionie i poziomie ma znaczenie tylko w przypadku wybrania samoczynnego przewijania. Zmniejszanie szybkości wprowadza dodatkowe opóźnienia względem ruchu myszy, co pozwala na eliminację przypadkowych drżeń, wahnięć itp. Ustawienie szybkości zależy też od używanej myszy. W przypadku myszki nagłownej wskazane jest używanie szybkości maksymalnych, aby uzyskać odpowiednie przemieszczenia przy jak najmniejszych wychyleniach głowy.
- ✓ Przytrzymanie nad obszarem klawiszy - zaznaczenie pozwala na użycie prawego przycisku klawisza *Enter* w celu ograniczenia ruchu kursora do pola klawiszy. Brak zaznaczenia powoduje, że prawy przycisk *Enter* pozostaje nieaktywny (zabezpiecza to przed przypadkowym włączeniem ograniczenia, jeśli ktoś nie chce go stosować).
- ✓ Pokazywanie opisu klawiszy - zaznaczenie powoduje przy każdorazowym wybraniu klawisza wyświetlenie skróconego opisu jego działania na belce tytułowej okna.

Działanie klawiszy regulują opcje:

- ✗ *Opóźnienie powtarzania* - określa jak szybko po przytrzymaniu wciśniętego klawisza nastąpi jego autopowtarzanie.
- ✗ *Szybkość powtarzania klawisza* - określa częstotliwość autopowtarzania po jego rozpoczęciu.

Należy zauważyć, że ustawienia te dotyczą wyłącznie programu klawiatury i nie mają nic wspólnego z podobnymi systemowymi nastawami dla klawiatury sprzętowej.

- ✗ *Opcja dodatkowa* (rysunek pochodzi z nieco starszej wersji) zaznaczenie uaktywnia dźwiękowy sygnał naciśnięcia klawisza (wymagana karta dźwiękowa).

Kolory używane w programie:

- ✗ Dla okienka edycji można wybrać jeden z 5 zestawów kolorów tła/tekst. Zrezygnowano z pełnej dowolności ustawiania kolorów ze względu na fakt, że stosunkowo nieliczne zestawienia zapewniają właściwy kontrast i dobrą widoczność kursora tekstowego.

- ✗ Dla pola klawiszy można regulować jasność. Jest to przydatne ze względu na używanie różnych i różnie wyregulowanych monitorów. Płynność regulacji zależy od ustawień karty graficznej. Użycie trybu HiColor (paleta 16-bitowa) zapewnia dobry efekt, natomiast uproszczona paleta 256 kolorów pozwala je-

dynie na dwustopniowe podświetlanie (ciemniej - jaśniej).

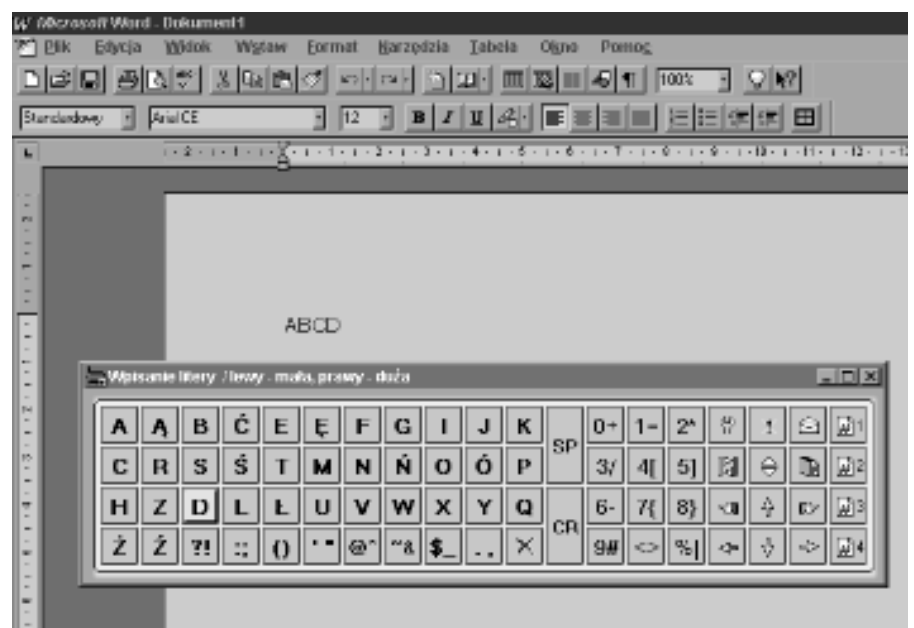
### Układy klawiatury

Podstawowy układ graficzny klawiatury jest stały. Natomiast można wybrać różne rozmieszczenia liter na części alfabetycznej klawiatury - co jest najbardziej istotne dla sprawności pisania i zgodności z własnymi przyzwyczajeniami. Kilka układów jest przygotowanych w programie: alfabetyczny, alfabetyczny ciągły (zgodny z ruchem kursora wzdłuż wierszy - pierwszy wiersz od lewej do prawej, drugi od prawej do lewej itd.), tradycyjny QWERTY (oczywiście z różnicami wynikającymi z obecności „polskich“ klawiszy), wspomagający pisanie członów dwuliterowych (cz, sz itd. - najczęściej występujące zestawienia liter są umieszczone w pobliżu siebie).

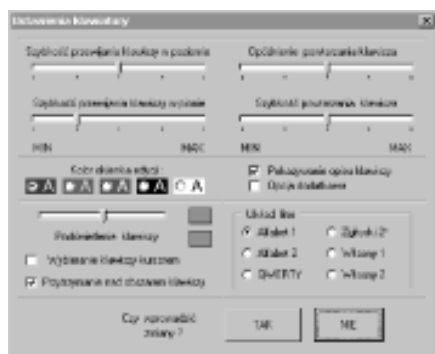
Wprowadzone ustawienia zatwierdzamy przyciskiem. W pełnej wersji instalacyjnej programu są one wtedy zapamiętane w rejestrze systemowym Windows. Skrócona wersja demonstracyjna nie daje możliwości zapamiętania ustawień - o czym użytkownik jest uprzedzany odpowiednim komunikatem.

### Przygotowanie własnego układu liter na klawiaturze

Do trybu konfigurowania klawiszy przechodzimy z podstawowego trybu edycji (z widocznym własnym oknem edycyjnym) poprzez



Rys. 4. Współpraca z zewnętrznym oknem przy klawiaturze przestanianej.



Rys. 5. Okienko konfiguracyjne.

funkcją F10. Lewy i prawy przycisk klawisza F10 wywołuje odpowiednio pierwszy i drugi układ liter. Na tle okna edycyjnego zostaje wyświetlony panel konfiguracji liter (rys. 6). Jednocześnie zmienia się nieco działanie klawiszy na klawiaturze. Prawym przyciskiem wybieramy z klawiatury klawisz do zmiany litery. Zostaje on zaznaczony w panelu (jak widać na rysunku - wybranie klawisza „S” spowodowało podświetlenie odpowiedniego pola w panelu). Wtedy naciskanie klawiszy klawiatury lewym przyciskiem spowoduje przepisanie odpowiedniej litery na zaznaczone miejsce.

Ustawianie układu klawiatury nie zostało wyposażone (jako że odbywa się sporadycznie) w mechanizmy kontroli poprawności. Zatem do użytkownika należy dopilnowanie, aby nowy układ zawierał wszystkie potrzebne litery bez braków i powtórzeń.

Początkowe (po instalacji) ustawienia pierwszego i drugiego układu użytkownika odpowiadają układowi alfabetycznemu. Wprowadzone własne zmiany zatwierdzamy przyciskiem „Zmień”. Podobnie jak przy konfiguracji nie ma możliwości zapamiętania ustawionej konfiguracji w wersji demonstracyjnej programu.



Rys. 6. Panel ustawiania własnego rozkładu liter na klawiaturze.

## Używanie dodatkowych zakładek tekstowych

Edycja tekstu prawie zawsze wymaga wycinania, wstawiania i przenoszenia słów, zdań czy akapitów. Zazwyczaj służy do tego schowek systemowy, jednakże często jest to niewystarczające. Każdy nowy zapis do schowka kasuje bowiem poprzednią zawartość, zaś nietrudno wyobrazić sobie sytuację, że chcemy posługiwać się kilkoma powtarzalnymi fragmentami tekstu jednocześnie. Istnieje wprawdzie wiele programów rozszerzających możliwości schowka, jednak obsługa dodatkowej aplikacji byłaby niewątpliwie poważnym utrudnieniem dla osoby korzystającej z myszy nagłownej. Dlatego wprowadzono cztery dodatkowe schowki - zakładki tekstowe obsługiwane w prosty sposób bezpośrednio z ekranowej klawiatury. Służy to tego 5 oddzielnych klawiszy. Cztery klawisze w skrajnej prawej kolumnie (rys. 1) pozwalają (w zależności od użycia lewego lub prawego przycisku) na przepisanie do danej zakładki zaznaczonego (podświetlonego) fragmentu z okna edycyjnego oraz na wstawienie zawartości zakładki w miejsce zgodne z pozycją kursora tekstowego. Podobnie jak w schowku systemowym, każdy następny wpis do zakładki kasuje poprzednią zawartość. Dlatego, aby uniknąć przypadkowego kasowania, zajętość zakładki (obecność w niej jakiegoś tekstu) jest sygnalizowana podświetleniem pola klawisza zakładki na czerwono.

Piąty klawisz służy do oglądania zawartości zakładek, gdyż na ogół bardzo szybko zapomnimy, co w której umieściliśmy. Przełączamy kolejnymi kliknięciami widoki zakładek - aktualnie oglądana ma pole klawisza podświetlone na żółto. Oprócz odczytania zawartości możemy ją normalnie edytować: wprowadzić poprawki lub całkowicie skasować dla zwolnienia zakładki. Jedynym ograniczeniem jest brak możliwości przenoszenia tekstu bezpośrednio pomiędzy zakładkami.

## Wymagania sprzętowe i instalacja programu

Wymogi sprzętowe nie są wygórowane. Wystarczy dowolny komputer z systemem operacyjnym

Microsoft Windows 95 lub 98. Jak wspomniano wcześniej, nieco lepsze wyposażenie (karta dźwiękowa, tryb Hi-color karty graficznej) zapewnia więcej możliwości, ale nie jest bezwzględnie konieczne. Monitor może pracować w podstawowej rozdzielczości VGA 640x480. Większe rozdzielczości umożliwiają łatwiejszą manipulację otwartymi oknami na pulpicie, ale z kolei utrudniają posługiwanie się myszą nagłowną. Z zebranych opinii użytkowników wynika, że maksymalną rozdzielczością ekranu będzie 800x600 pikseli.

Aplikację umieszczamy w systemie w standardowy sposób, uruchamiając z dyskiety program instalacyjny *setup.exe*. Skopiuje on potrzebne pliki do wybranego przez użytkownika folderu (domyślnie *c:\Program Files\Avt\Klawiatura*), dokona niezbędnych wpisów w rejestrze systemowym i umieści potrzebne pozycje w menu *Start/Programy*. Jeśli zechcemy aplikację usunąć, nie kasujemy zawartości foldera z programem, ale użyjemy pozycji menu *Usuń program* (albo okna *Dodaj/Usuń Programy* w Panelu Sterowania).

Dla zapoznania się z wersją demonstracyjną wystarczy skopiować do wybranego folderu pliki *keys.exe* i *fohook.dll* (dostępne również na stronie [www.ep.com.pl](http://www.ep.com.pl)). Ta operacja nie zmienia zawartości rejestru Windows i w razie potrzeby wystarczy zwykle skasowanie plików.

Aplikacja jest w założeniu przewidziana do współpracy z myszką nagłowną, ale podłączenie tej myszy nie jest warunkiem instalacji i uruchomienia - można używać programu z innymi typami manipulatorów, w zależności od możliwości i sprawności rąk.

**Jerzy Szczesiul, AVT**  
[jerzy.szczesiul@ep.com.pl](mailto:jerzy.szczesiul@ep.com.pl)

*Program wirtualnej klawiatury do myszki dla niepełnosprawnych jest dostępny w Internecie pod adresem: [www.ep.com.pl/ftp/mysz.exe](http://www.ep.com.pl/ftp/mysz.exe) oraz na płycie CD-EP6/2000.*

*Autor pragnie przekazać serdeczne podziękowania koledze Andrzejowi Abraszewskiemu, inicjatorowi powstania opisywanego programu, a także wszystkim osobom testującym wersję próbną, których uwagi bardzo pomogły w ustaleniu docelowych funkcji i działania aplikacji.*